

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Erik Cvetek

**Aplikativna podpora upravljanju in
vzdrževanju modelov s teflonskim
premazom**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2015

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomskega dela razvijte aplikacijo za upravljanje z modeli, ki se uporabljajo v proizvodnem procesu izdelave avtomobilskih filtrov. Aplikacija naj bo integrirana v obstoječ informacijski ERP sistem. Rešitev naj omogoča enostavno upravljanje z modeli, hitro elektronsko kreacijo naročilnice za servis teflonskih oblog, evidenco prevzema modelov nazaj v podjetje, evidenco preventivnega vzdrževanja in evidenco pregledov na zahtevo ter grafični pregled nad trenutnim stanjem modelov. Pri implementaciji se posebno osredotočite na uporabniku intuitivno ter časovno nezahtevno uporabo aplikacije.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Erik Cvetek sem avtor diplomskega dela z naslovom:

Aplikativna podpora upravljanju in vzdrževanju modelov s teflonskim premazom

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luka Šajna,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 3. julija 2015

Podpis avtorja:

Zahvaljujem se vsem svojim bližnjim, ki so me tekom študija podpirali in spodbujali. Posebno se zahvaljujem mentorju doc. dr. Luka Šajnu za pomoč ter nasvete pri izdelavi diplomskega dela.

Zahvaljujem se tudi podjetju Sogefi Filtration d. o. o. za strojno in programsko opremo, ki sem jo potreboval za izvedbo praktičnega dela diplomske naloge. Posebna zahvala gre tudi Alešu Bertalaniču, Stephanu Axusu, Maximu Raddeju in Leopoldu Šmitu za spodbudo ter tako organizacijsko kot idejno podporo.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Tehnologije in orodja	5
2.1	Uporabljene tehnologije	5
2.1.1	ASP.NET	5
2.1.2	HTML	6
2.1.3	CSS	6
2.1.4	C#	6
2.1.5	SQL	7
2.2	Razvojna okolja	7
2.2.1	Microsoft Visual Studio	7
2.2.2	SQL Server Management Studio	8
2.3	Uporabljene knjižnice	8
2.3.1	Bootstrap	8
2.3.2	JustGage	8
3	Načrtovanje in razvoj	9
3.1	Primeri uporabe spletne aplikacije	9
3.1.1	Upravljanje s seti z modeli	9
3.1.2	Naročilnice	10

KAZALO

3.1.3	Prezemnice	11
3.1.4	Pregledi	11
3.1.5	Pregled statusa življenja	12
3.1.6	Zgodovina uporabe	12
3.2	Primeri uporabe namizne aplikacije	12
3.2.1	Sporočanje	13
3.3	Načrtovanje podatkovne baze	13
3.4	Implementacija spletne aplikacije	15
3.4.1	Upravljanje s seti modelov	15
3.4.2	Izračun stanja življenja modelov	17
3.4.3	Prikaz stanja življenja modelov	19
3.4.4	Detajli seta modelov	19
3.4.5	Naročilnice	22
3.4.6	Prezemnice	22
3.4.7	Pregledi	24
3.4.8	Poročila	25
3.4.9	Dnevnik uporabe	25
3.5	Implementacija namizne aplikacije	26
3.5.1	Prijava	26
3.5.2	Glavno okno	28
3.5.3	Zgodovina	28
4	Sklepne ugotovitve	31
	Literatura	33
	Priloga A	37
	Priloga B	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
ERP	Enterprise Resource Planning	informacijski sistemi za upravljanje z viri podjetja
CSS	Cascading Style Sheets	kaskadne slogovne podloge
HTML	Hyper Text Markup Language	označevalni jezik za nadbesedila
SQL	Structured Query Language	strukturirani povpraševalni jezik
MSSQL	Microsoft SQL Server	Microsoftov strežnik SQL
SSRS	SQL Server Reporting Services	storitve za poročanje strežnika SQL
XML	Extensible Markup Language	razširljiv označevalni jezik
IIS	Internet Information Services	informacijske internetne storitve

Povzetek

Cilj diplomskega dela je bil izdelati aplikacijo za upravljanje in vzdrževanje orodja s teflonskim premazom (setov z modeli), ki se uporabljajo v avtomobilski proizvodnji kabinskih, zračnih in oljnih filtrov. Glavni namen aplikacije je omogočiti pregled nad obrabo teflonskih oblog, pravočasno preventivno vzdrževanje in pregledovanje orodja ter kreacija elektronske naročilnice za servis teflonskih oblog ob izrabi. S temi nameni je bila ustvarjena aplikacija, ki transparentno s svojim delovanjem pokriva vidike financ, logistike in kakovosti.

V diplomskem delu je opisano vse od glavnih uporabljenih tehnologij do knjižnic, programske opreme, idejne zasnove, načrtovanja, razvoja in integracije ter predaja aplikacije v končno produkcijsko okolje uporabnikov. Na koncu se bomo dotaknili tudi rezultatov ter možnost nadaljnjega razvoja in primernost širše uporabe.

Ključne besede: aplikacija, teflonski premaz, modeli, proizvodnja, teflonizacija, vzdrževanje, upravljanje.

Abstract

The objective of my thesis is to make an application for managing and maintaining tools with Teflon coating (mouldsets), which are used in automotive industry of cabin, round air and oil filters. Main purpose of the application is to enable the overview with end of life status of Teflon coating, preventive maintenance, checkup of tools and in the end, creation of electronic order for refurbishing the Teflon coating before it gets used up. The application was designed to meet the criteria described above and transparently cover the aspects of finance, logistic and quality departments.

In my thesis I will describe the whole process from used technologies to libraries, software, concept, planning, developing and integration followed by handover of application to production environment of final users. In the end we will check the final results and scratch the surface of possible further development and capability of wider common use.

Keywords: application, teflon coating, moulds, production, teflonisation, maintenance, management.

Poglavje 1

Uvod

V zadnjem desetletju nas je izredno hiter razvoj na področju računalništva prisilil, da smo začeli opuščati papirnate medije in se osredotočili na elektronsko hrambo podatkov - tudi zaradi kvalitetnejše in predvsem lažje analize hranjenih podatkov. Ob tem prehodu je seveda nastal neizbežen problem: Kako najučinkoviteje in kvalitetno shraniti podatke, da bodo le-ti v prihodnosti dostopni tudi novejšim tehnologijami za analitično obdelavo ter poročanja.

Razcvet je tako kot računalniško področje doživela tudi mehanizacija v proizvodnih podjetjih. Ker je čas denar, je v proizvodnjah zelo malo prostora za manevriranje s pisanjem poročil, čeprav so ta konec dneva najpomembnejša. Zaradi pomena poročil so se nekatera podjetja po svetu odločila za implementacijo obstoječih sistemov, nekatera pa za razvoj lastnih. V podjetju Sogefi Filtration d. o. o. so bile zahteve proizvodnje zelo specifične, zato so se odločili za razvoj lastnega informacijskega ERP sistema in bazirali pregled ter učinkovitost proizvodnje z dnevnimi sestanki ter poročanjem, katerih podlaga so bila ročno izdelana poročila na podlagi papirnatih poročil posameznih izmen in posameznih proizvodnih linij.

Ker je proces poročanja obsegal precejšen delež delovnika, smo se odločili, da združimo podatkovne baze različnih informacijskih ERP sistemov in izdelamo aplikacijo, ki bo pomagala, opravljala delo in opozarjala odgovorne o

stanju iztrošenosti orodij na proizvodnih linijah. Tako bodo zaposleni lahko svoj čas usmerili v druge bolj pomembne procese proizvodnega življenja.

Na podlagi vseh znanih dejstev smo se odločili, da izdelamo aplikacijo, ki se osredotoča samo na orodja s teflonskim premazom, saj so ta največji in najpomembnejši del proizvodne kakovosti (sliki 1.1 in 1.2). Aplikacija v svojem obsegu poročanja posledično zajema planerje proizvodnje, kontrolorje kakovosti in finančne analitike. Uporabnikom vzdrževanja poleg pregleda nad trenutnim statusom modelov aplikacija nudi tudi časovni okvir planiranja servisov in učinkovitejše manipuliranje s finančnimi sredstvi, kar pa je tudi najpomembnejši vidik uspešnosti podjetja. Tako omejimo tudi človeški faktor pozabljivosti in se izognemo prepozmem servisu orodja, kar običajno pripelje do slabe kvalitete izdelave, izmeta in v najslabšem primeru izgube posla.

Celotna aplikacija je zaradi združevanja najzanesljivejših virov informacij razdeljena na dva dela, in sicer na spletni portal, ki je namenjen vzdrževalcem in planerjem, in namizno aplikacijo, ki je namenjena vsem proizvodnim delavcem, ki imajo dostop do informacijskega ERP sistema na računalnikih v proizvodnji. Preko te aplikacije proizvodni delavci sporočajo opazke in zahteve za pregled orodja delavcem v vzdrževanju.

V prvem delu diplomske naloge bomo na kratko pregledali uporabljene tehnologije, programska orodja in vključene knjižnice, ki smo jih uporabili pri obeh delih aplikacije. Nato pa se bomo v drugem in tretjem delu osredotočili na problem, platformo in razvoj spletnega portala ter namizne aplikacije, na kratko bomo pregledali osnovni tok delovanja obeh aplikacij.



Slika 1.1: Prikaz odprtega modela z vidno teflonsko oblogo v zeleni barvi.



Slika 1.2: Uporaba modelov v proizvodnem procesu.

Poglavje 2

Tehnologije in orodja

Ker je bil naš cilj razviti spletno aplikacijo na intranet ravni, smo se odločili, da izberemo programski jezik ASP.NET [23, 24] v kombinaciji s HTML 4 in 5 [2, 3] za prikaz dinamičnih objektov ter komunikacijo s strežnikom IIS. Za vizualno podobo smo poskrbeli z uporabo programskega jezika CSS. Za vsebinski del spletne aplikacije in razvoj namizne komunikacijske aplikacije smo uporabili programski jezik C# [7, 8]. Ker je bil obstoječ informacijski ERP sistem že baziran na podatkovni bazi MSSQL, smo se odločili, da tu ne bomo vpeljevali dodatnih sprememb in bomo uporabili že obstoječo podatkovno bazo. Razvoj tako spletnega kot namiznega dela aplikacije je v celoti potekal v Microsoftovem Visual Studiju [15, 16], za načrtovanje in izdelavo podatkovne baze pa smo uporabili SQL Server Management Studio. Pomagali smo si tudi z orodji za razvijalce v spletnem brskalniku Google Chrome.

2.1 Uporabljene tehnologije

2.1.1 ASP.NET

ASP.NET [21, 22] spada med relativno mlajša programska ogrodja. Prvič se je na trgu pojavil januarja 2002 v paketu z .NET Framework kot naslednik ASP. Gre za odprtokodno strežniško orientirano programsko ogrodje za izde-

lavo dinamičnih spletnih strani. Ker je ASP.NET ogrodje zgrajeno z uporabo CLR prevajalnika, podpira in je zmožno prevesti kateri koli programski jezik v .NET zbirki jezikov (Visual C++, Visual C#, Visual Basic, Visual F# ipd).

2.1.2 HTML

Programski jezik HTML [1] je najosnovnejši in najbolj razširjen označevalni jezik za izdelavo enostavnih ali kompleksnih spletnih strani. Uporablja se za definicijo strukturnega in semantičnega dela spletnih dokumentov, ki jih spletni strežnik interpretira in prevede v obliko, ki je primerna za prikaz na naših ekranih z uporabo spletnih brskalnikov. Začetki programskega jezika segajo daleč v leto 1991, ko je bila izdana prva verzija, v kateri je bilo definiranih le nekaj osnovnih značk (značka povezave, naslovi, sezname, ipd), ki so se še danes obdržale kot del osnovnih gradnikov v najnovejših verzijah programskega jezika HTML.

2.1.3 CSS

Slogovni programski jezik CSS [4, 5] skrbi za slogovno predstavitev spletnih dokumentov. S podlogami, ki so predstavljene v programskem jeziku CSS, definiramo, kako naj se elementi jezika HTML prikazujejo na spletni strani. Z njimi lahko predstavimo attribute tako na vizualnem (barve, velikosti, odmiki, poravnane ipd) kot na odzivnem področju (prelet z miško ipd). Glavno bistvo slogovnega jezika CSS je poenostavitev kode z ločitvijo spletnega dokumenta na strukturni in predstavitveni nivo. Tako se izognemo nepotrebnemu ponavljanju programske kode, poskrbimo za preglednost in omogočimo uporabo iste predstavitvene predloge več spletnim dokumentom.

2.1.4 C#

Začetki programskega jezika C# [6] segajo v leto 1999, v čas razvoja knjižnic .NET Framework, takrat še s prvotnim imenom Cool (C-like Object Oriented

Language). Leto kasneje je bil v javnost izdan s paketom knjižnic .NET Framework preimenovan v C#. Programski jezik je objektno orientiran, zasnovan s cilji enostavnosti, modernosti in splošne uporabe. Določen je s standardi ECMA in ISO.

2.1.5 SQL

Povpraševalni jezik SQL [18, 19] je najbolj razširjen ter standardiziran odprt kodni jezik za delo s podatkovnimi bazami. Uporablja se za izdelavo poizvedb oziroma delo s podatki nad podatkovnimi bazami s sintakso, ki je zelo podobna ukazom v naravnem jeziku. Začetki jezika segajo v leto 1970, najdemo jih pod prvotnim imenom SEQUEL, kasneje pa je bil jezik preimenovan v SQL. V javnost je bil izdan leta 1986, podprt je bil s standardoma ANSI in ISO. Od leta 1986 dalje je bilo v javnost izdanih več različnih dialektov povpraševalnega jezika SQL.

2.2 Razvojna okolja

2.2.1 Microsoft Visual Studio

Microsoft Visual Studio [13, 14] je razvojno okolje, ki je namenjeno razvoju aplikacij za operacijske sisteme Microsoft Windows, razvoju spletnih aplikacij, spletnih strani in storitev. Zelo pomembna kvaliteta ogrodja za oblikovanje kode je IntelliSense komponenta, ki skrbi za avtomatsko dokončevanje pri pisanju kode. Razvojno okolje vsebuje tudi oblikovalca spletnih dokumentov, razredov in podatkovnih baz ter podpira pisanje, prevajanje, poganjanje in razhroščevanje kode vseh razvojnih jezikov paketa .NET Framework (C, C++, VB.NET, F# in C#) ter spletnih razvojnih jezikov (HTML, XML, CSS in JavaScript). Razvojno okolje je javnosti na voljo v več različicah, med njimi je tudi brezplačna Express različica.

2.2.2 SQL Server Management Studio

SQL Server Management Studio [20] je aplikacija, ki je namenjena upravljanju in administraciji komponent podatkovne baze na Microsoft strežniku SQL. Izdana je bila skupaj z različico podatkovne baze Microsoft strežnika SQL 2005. Orodje obsega tudi poganjanje poizvedb in delo z elementi podatkovnih baz.

2.3 Uporabljene knjižnice

2.3.1 Bootstrap

Bootstrap [11, 12] je odprtokodno brezplačno ogrodje za izdelavo dinamičnih spletnih strani. Paket vsebuje predpripravljene slogovne predstave (CSS podloge) in označevalne elemente (HTML elementi), ki jih lahko uporabimo pri razvoju spletnih dokumentov za doseganje dinamičnosti oblikovnega prikaza spletnih dokumentov na namiznih, mobilnih ter tabličnih spletnih brskalnikih. Z uporabo knjižnice Bootstrap je razvita spletna aplikacija vedno prikazana in prilagojena širinami ekrana na različnih napravah.

2.3.2 JustGage

JustGage [17] je odprtokodna ter brezplačna knjižnica vektorskih grafičnih objektov za spletne dokumente. Knjižnica vsebuje animirane števec za eleganten in jasen prikaz številčnih vrednosti s pripadajočo minimalno in maksimalno mejo. Knjižnica je razvita v programskem jeziku JavaScript [9, 10] z uporabo osnovnih označevalnih elementov in uporabo umerljive vektorske grafike, je združljiva z vsemi najbolj razširjenimi spletnimi brskalniki (Internet Explorer, Opera, Google Chrome, Firefox, Android, Safari ipd).

Poglavje 3

Načrtovanje in razvoj

Načrtovanje in razvoj aplikacije smo razdelili na dva segmenta. V prvem delu smo analizirali zahteve v sklopu spletne aplikacije, v drugem delu pa zahteve namizne aplikacije. Na podlagi analiz smo nato opredelili primere uporabe, s katerimi smo si pomagali pri načrtovanju podatkovne baze in pri implementaciji rešitve.

3.1 Primeri uporabe spletne aplikacije

Ker je ciljno okolje omrežje intranet in s tem uporabniki spletne aplikacije izključno znotraj podjetja, nam ni bilo potrebno razčlenjevati uporabnikov na nivoje dostopa. Vsi uporabniki, tako navadni uporabniki kot administratorji, imajo enak nivo dostopa. Primere uporabe smo zaradi lažje inkrementalne implementacije razčlenili na sedem glavnih segmentov delovanja, in sicer: *pregled stanja*, *modeli*, *naročilnice*, *prevzemnice*, *pregledi*, *poročila* in *zgodovina akcij*. Dostop do aplikacije je voden preko članstva namenske skupine na domenskem aktivnem imeniku.

3.1.1 Upravljanje s seti z modeli

Uporabniki morajo imeti možnost ustvarjanja novih setov z modeli ter brisanja in deaktiviranja neaktivnih setov. Prav tako morajo imeti možnost

upravljanja z vsemi atributi vezanimi na set modelov. Vsak set mora imeti dve glavni meji življenja, in sicer meja opozorila ter kritična meja. Ti dve meji morata biti razdeljeni na mejo posameznega modela in mejo skupka vseh modelov v setu. Meja opozorila naj se uporablja pri preventivnem obveščanju preko elektronskih sporočil, kritična meja pa je končna oziroma maksimalna meja, pri kateri se modeli še lahko uporabljajo na proizvodnih trakovih. Vsak set modelov mora biti vezan na eno ali več tehničnih kod, ki so uporabljene v proizvodnji, kar pomeni, da se mora pri štetju izdelanih kosov upoštevati vse tehnične kode, ki so vezane na set modelov. Vsak set mora biti vsebovan pod točno eno proizvodno linijo, kar pomeni, da se pri štetju izdelanih kosov upoštevajo samo izdelki tehničnih kod, ki so izdelane na izbrani proizvodni liniji, vse ostalo se ignorira.

Uporabniki morajo imeti na voljo tudi enostaven iskalnik za iskanje setov z modeli po njihovih imenih.

3.1.2 Naročilnice

Uporabniki morajo imeti možnost ustvarjanja novih naročilnic in po potrebi tudi brisanja le-teh. Ko uporabnik želi kreirati novo naročilnico za servis teflonske obloge, mora najprej izpolniti osnovne podatke naročilnice (zaporedna številka naročilnice, datum in kraj izdaje, dobavitelj in stroškovno mesto). S tem korakom se v aplikaciji ustvari začasna naročilnica. V drugem koraku mora sistem dovoliti izbor vseh aktivnih setov z modeli, ki so trenutno na lokaciji podjetja in so na voljo za pošiljanje na servis teflonskih oblog. Naročilnica lahko vsebuje enega ali več modelov. V naslednjem koraku mora aplikacija izbrane sete z modeli razdeliti na spodnje in zgornje dele ter uporabniku dovoliti urejanje predhodno izpolnjenega obrazca. Pri tem koraku mora uporabnik potrditi pregled vnesenih količin in s to akcijo prevzeti odgovornost za možne napake pri vnosu. V zadnjem koraku aplikacija uporabniku ponudi tiskalniku prijazno naročilnico, ki je pripravljena za tiskanje in podpisovanje. Ko je naročilnica natisnjena, aplikacija avtomatsko spremeni stanje naročilnice v *ZAKLJUČENO* ter ponastavi števec življenja

za vse sete z modeli na naročilnici. Če uporabnik prekine proces izpolnjevanja naročilnice pred tiskanjem, si mora sistem zapomniti vse že vnesene parametre in na seznamu naročilnico prikazati kot nedokončan dokument.

Uporabnik mora imeti možnost brisanja tako dokončanih kot nedokončanih naročilnic, ne sme pa mu biti dovoljeno izbrisati naročilnico, ki ima nase že vezano prevzemnico. Aplikacija mora poskrbeti za pravilno zaklepanje dokumentov ter, v primeru brisanja, obnovitev števcov življenja nazaj na stanje pred kreacijo naročilnice.

Aplikacija mora nuditi vodenje in prikaz zgodovine vseh naročilnic.

3.1.3 Prezemnice

Aplikacija mora uporabniku omogočati pregled nad vsemi odprtimi naročilnicami, ki še nimajo vezave na prevzemnico. Na podlagi tega prikaza mora aplikacija uporabniku omogočiti kreacijo prevzemnice. Uporabniku mora biti omogočen zapis komentarja pri izdelavi prevzemnice, potrditi mora število prevzetih modelov in s tem prevzeti odgovornost za možne napake pri prevzemu. Ko je prevzemnica vnesena v aplikacijo, se mora naročilnica, nad katero je bil kreiran prevzem, zakleniti za brisanje.

Aplikacija mora nuditi vodenje in prikaz zgodovine vseh prevzemnic, nuditi pa mora tudi opcijo brisanja le-teh. Ko uporabnik izbriše prevzemnico, sprostí vezano naročilnico in omogoči akcijo njenega izbrisa.

3.1.4 Pregledi

Aplikacija mora uporabnikom v vzdrževanju omogočati pregled vseh sporočil iz namizne aplikacije, ki je nameščena na proizvodnih računalnikih, ter kreacijo pregleda na podlagi le-teh. Pri vnašanju pregleda mora imeti uporabnik možnost vpisa komentarja. Pred potrditvijo mora potrditi opravljen pregled ter tako prevzeti odgovornost za opravljeno delo. Aplikacija mora uporabniku omogočati tudi vnašanje preventivnih pregledov brez vezave na sporočila iz namizne aplikacije.

Aplikacija mora nuditi vodenje in prikaz zgodovine vseh pregledov, nuditi pa mora tudi opcijo njihovega brisanja. Ko uporabnik izbriše pregled, ki je vezan na sporočilo, se mora le-to sprostiti in ponovno prikazati na seznamu čakajočih sporočil za pregled. Šele ko uporabnik izbriše sporočilo, se to umakne iz seznama.

3.1.5 Pregled statusa življenja

Aplikacija mora uporabniku omogočati enostaven pregled nad vsemi aktivnimi seti z modeli ter njihovo stanje življenja, ki je izračunano na podlagi mej življenja, ki so nastavljene v atributih. Prikaz naj bo jasen in v grafični obliki, omogočati mora tudi filtriranje po stanjih in po lokacijah.

Uporabniki morajo imeti možnost detajlnega pregleda seta z modeli. Detajlni pregled mora vsebovati pregled nad celotnimi atributi seta, pregled nad tekočim številom uporab modelov in pregled nad celotno zgodovino akcij opravljenih nad izbranim setom z modeli.

3.1.6 Zgodovina uporabe

Aplikacija mora hraniti ter omogočati pregled celotne zgodovine akcij. Zgodovina mora biti deljena po različnih sekcijah aplikacije in različnih opravljenih operacijah znotraj vsake sekcije. V zgodovini mora biti razvidno, kdaj je bila akcija izvedena ter kdo jo je izvedel, prav tako pa mora biti razvidna tudi teža akcije. Zgodovina mora biti vidna vsem uporabnikom.

3.2 Primeri uporabe namizne aplikacije

V sklop celote spada tudi namizna aplikacija, ki je nameščena na vse računalnike v proizvodnem okolju. Dostop do aplikacije je omejen izključno na proizvodne uporabnike, ki imajo uporabniški račun za uporabo obstoječega informacijskega ERP sistema. Za prijavo se uporablja enake poverilnice kot pri prijavi v informacijski ERP sistem.

3.2.1 Sporočanje

Namizna aplikacija mora biti zaklenjena s prijavo in omogočati dostop samo določenim proizvodnim uporabnikom. Pri prijavi mora imeti aplikacija že prednastavljeno proizvodno linijo (lokacijo) oziroma omogočati izbor v primeru, da je računalnik v skupni uporabi več proizvodnih linij.

V namizni aplikaciji mora uporabnik imeti možnost izbora imen modelov, ki so vezani na izbrano lokacijo, ki je določena pri prijavi. Uporabnik mora vnesti besedilo sporočila, če želi izvesti akcijo sporočanja v oddelek vzdrževalnih pregledov. Prav tako mora imeti možnost pregleda zgodovine vseh vnesenih sporočil pod prijavljeno lokacijo ter njihova stanja (poslano in pregledano).

3.3 Načrtovanje podatkovne baze

Po opravljeni analizi primerov uporabe smo začeli načrtovati podatkovno bazo. Baza vsebuje dvanajst tabel uporabljenih in ustvarjenih izključno za namene spletne in namizne aplikacije ter šest tabel skupne narave z obstoječim informacijskim ERP sistemom. Na sliki B.1 so prikazane tabele, ki so glede na funkcionalnost razdeljene na šest logičnih enot.

- Logična enota *seti modelov* je sestavljena iz dveh tabel:

- TEFL_MOULDSETS,
- TEFL_CODE_LINKS.

V tabeli TEFL_MOULDSETS se hranijo osnovni podatki o setih z modeli, v tabeli TEFL_CODE_LINKS pa njihove povezave na tehnične kode. Set modelov ima lahko več povezav na različne tehnične kode.

- Logična enota *naročilnice* je sestavljena iz petih tabel:

- TEFL_ORDERS,
- TEFL_ORDERS_MOULDS,

- TEFL_ORDERS_QTYTRACE,
- TEFL_ORDERS_SEQ,
- TEFL_ORDERS_WELDING.

V tabeli TEFL_ORDERS se hranijo osnovni podatki o posamezni naročilnici, v tabeli TEFL_ORDERS_MOULDS pa nanjo povezani seti modelov. Naročilnica lahko vsebuje več povezav na različne sete modelov. V tabeli TEFL_ORDERS_WELDING se hrani število modelov, ki jih je treba variti. Vsaka naročilnica ima lahko samo eno postavko varjenja, v njej pa seštevek vseh možnih modelov, ki potrebujejo varjenje. To število je le groba ocena, ki pa dejansko lahko variira po opravljenem strokovnem pregledu pri ponudniku. Tabela TEFL_ORDERS_QTYTRACE hrani število uporab seta modelov, ob katerih je bil poslan na servis teflonskih oblog. Vsaka naročilnica ima točno enako število zapisov v tabeli TEFL_ORDERS_QTYTRACE, kot je število setov z modeli vezanih na naročilnico. V tabeli TEFL_ORDERS_SEQ pa se hrani zadnja zaporedna številka izdane naročilnice v tekočem letu.

Logična enota je relacijsko povezana tudi z logično enoto *modeli* ter lahko vsebuje točno toliko povezav, kolikor ima logična enota *naročilnice* setov z modeli, ki so vezani nanjo.

- Logična enota *prevzemnice* je sestavljena iz ene tabele:

- TEFL_RECIEVES.

V tabeli TEFL_RECIEVES se hranijo osnovni podatki o prevzemu seta modelov iz servisa teflonski oblog nazaj v prostore podjetja.

Logična enota je relacijsko povezana tudi z logično enoto *naročilnice* in ima lahko z njo točno eno povezavo.

- Logična enota *pregledi* je sestavljena iz treh tabel:

- TEFL_INSPECTIONS,

- TEFL_MESSAGES,
- TEFL_CLIENTS.

V tabeli `TEFL_MESSAGES` se hranijo sporočila, ki so poslana v spletno aplikacijo z uporabo namizne aplikacije za sporočanje o napakah. Tabela `TEFL_INSPECTIONS` hrani komentarje pregledov setov z modeli in ima lahko največ eno ali nobene povezave s tabelo `TEFL_MESSAGES`. V tabeli `TEFL_CLIENTS` se hranijo preddefinirani računalniki s fiksnimi povezavami na določeno proizvodno linijo oziroma na uporabnika.

Logična enota je relacijsko povezana tudi z logično enoto *modeli* in ima lahko z njo točno eno povezavo, in sicer z vsako od tabel `TEFL_INSPECTIONS` in `TEFL_MESSAGES`.

- Logična enota *zgodovina* je sestavljena iz ene tabele:

- `TEFL_ACTION_LOG`.

V tabeli `TEFL_ACTION_LOG` se hrani celotna zgodovina uporabe spletne aplikacije na uporabniški in aplikativni ravni.

Logična enota je na podlagi izvedenega tipa operacije relacijsko povezana z vsemi ostalimi logičnimi enotami.

3.4 Implementacija spletne aplikacije

Po izdelani podatkovni bazi smo se lotili implementacije. To smo razčlenili na dva dela, in sicer izdelava spletne aplikacije ter izdelava namizne aplikacije.

3.4.1 Upravljanje s seti modelov

Modeli oziroma seti modelov so ključnega pomena za delovanje celotne aplikacije. Upravljanje s seti, modeli ter kakovost vnesenih podatkov so ključni elementi vseh nadaljnjih izračunov, zato smo jim posvetili nekoliko več časa

TEFLONIZACIJA

EOL plošča

Seti modelov

Naročilnice

Prevzemnice

Pregledi

Poročila

Zgodovina

Pomoč

Upravljanje s seti modelov

Išči po imenu:

Išči

* Nov set modelov

☐ Pokaži izbrisano

ID	Ident	Črtna koda	Ime seta	Linija	Lokacija	Je aktiven			
185	6578	000132	41.40.028	P2	106	DA	Uredi	Izbriši	
101	8797	000245	41.40.091	P2	25	DA	Uredi	Izbriši	
3	7645	000326	41.40.107	P4	4,5	DA	Uredi	Izbriši	
4	6457	000749	41.40.108	P4	6	DA	Uredi	Izbriši	
124	4834	000988	41.40.169	P2	7	NE	Uredi	Izbriši	
126	8456	000175	41.40.171	P2	28	DA	Uredi	Izbriši	
11	7897	000989	41.40.195	P4	20	DA	Uredi	Izbriši	
145	3574	000983	41.40.258	P2	1	DA	Uredi	Izbriši	
219	9848	000545	41.40.288	P3	132	NE	Uredi	Izbriši	
220	3231	000777	41.40.288	P3	140	DA	Uredi	Izbriši	

Slika 3.1: Pregled seznama in urejanje setov z modeli.

in poskusili zgraditi uporabniku prijazen, enostaven in pregleden vmesnik za urejanje setov.

Na sliki 3.1 je prikazan osnovni pogled seznama vseh setov z modeli. V zgornjem delu se nahaja enostaven iskalnik za hitro iskanje setov po njihovih imenih. Poleg njega je gumb, ki uporabnika preusmeri na podstran za dodajanje novega seta modelov.

Seznam prikaza je zgrajen iz najbolj osnovnih ter najbolj pomembnih podatkov, ki uporabniku omogočajo hitro identifikacijo seta modelov. V desnem kotu seznama sta dve hitri bližnjici, ki uporabnika preusmerita na obrazec za urejanje ali na obrazec za brisanje. Skrajno desno se nahaja indikator, ki prikazuje ali je set modelov aktiven in prikazan na poročilih, ostalih izračunih, ali pa gre za neaktiven set modelov in se trenutno ne uporablja v proizvodnji ter ni prikazan pri nobenem izračunu v spletni aplikaciji (evidenčno upravljanje seta modelov). Poleg grafičnega indikatorja je status modela prikazan tudi z barvo, in sicer so s svetlo sivo označeni deaktivirani

seti modelov.

Na sliki 3.2 je prikazana stran za urejanje nastavitev seta modelov. V zgornjem delu se nahajajo osnovne nastavitve, kjer uporabnik vnese identifikacijske in opisne podatke o setu. Ker so določena polja obvezna, se pri shranjevanju označijo z zvezdico in indikatorjem o napaki pri vnosu podatkov.

V osrednjem delu, kjer se nahajajo *End od Life meje*, uporabnik vnese število izdelanih kosov, ki jih posamezen model prenese preden se sproži meja opozorila ali kritična meja. Pri kritični meji se modeli ob pravilno vnesenih mejah ne smejo več uporabljati v proizvodnem procesu. Meje se vedno vnašajo na nivoju posameznega modela in se preko parametra o številu modelov znotraj seta skupaj z vneseno mejo preračunajo na nivo celotnega seta modelov, kar uporabniku pomaga pri predstavi vnesenih meja.

V spodnjem delu se nahaja najpomembnejši del parametrov, in sicer *Povezave tehničnih kod*. V tem delu uporabnik izbere in na seznam vezanih tehničnih kod doda le tiste tehnične kode, pri katerih se set modelov uporablja v proizvodnem procesu. Na podlagi vezanih tehničnih kod in vezane proizvodne linije (proizvodni trak) se v spletni aplikaciji ustvari logična povezava z informacijskim ERP sistemom, v katerega se na proizvodnih računalnikih vnaša izdelane količine ter preračuna dejansko število uporab modelov ter njihovo stanje življenja.

3.4.2 Izračun stanja življenja modelov

Izračun stanja življenja modelov spada med najbolj kompleksne algoritme, ki so uporabljeni v celotnem projektu. Na izseku kode A.1 je prikazana poizvedba SQL, ki povezuje kar šest tabel na različnih nivojih, nekatere celo več kot enkrat.

Poizvedbi SQL moramo za pravilen prikaz podati dva parametra. Parameter `@p1` predstavlja filter proizvodne linije in ga tipično pridobimo iz spletnega obrazca v tekstovni obliki.

Parameter `@p2` predstavlja najnižjo datumsko mejo, ki je uporabljena pri

Osnovne nastavitve

ID	<input type="text" value="185"/>
Ime	<input type="text" value="41.40.028"/>
Ident	<input type="text" value="4140028"/>
Črtna koda	<input type="text" value="001587"/>
Linija	<input type="text" value="P2"/>
Lokacija	<input type="text" value="106"/>
Število modelov	<input type="text" value="75"/>
Velikost modelov	<input type="text" value="ENOJNI"/>
Je aktiven	<input checked="" type="checkbox"/>

End od Life meje

Opozorilo:	<input type="text" value="1000"/>	kosov na model	<input type="text" value="75000"/>	kosov na set
Kritično:	<input type="text" value="1200"/>	kosov na model	<input type="text" value="90000"/>	kosov na set

Povezave tehničnih kod

Proste tehnične kode

41.10.34

41.10.36

41.10.53

41.10.89

41.10.143

41.10.144

41.10.215

41.10.216

41.10.235

41.10.239

41.10.240

41.10.240A

>

<

Vezane tehnične kode

41.40.28

41.40.28

Slika 3.2: Prikaz stani za urejanje seta modelov.

izračunu izdelanih kosov v primeru, da set modelov še nikoli ni bil poslan na servis teflonskih oblog. Parameter `@p2` je statično kodiran v nastavitvah kot varovalka poizvedbe pred nepotrebno potratu časa za izračun nerelevantnih podatkov in je podan v obliki tipa *long*.

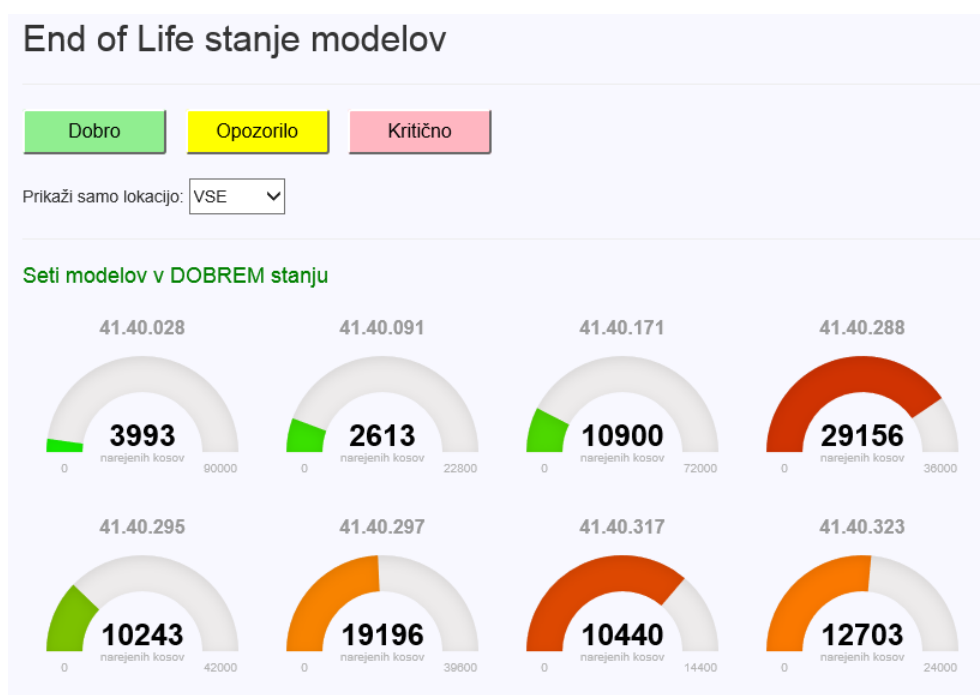
Poizvedba SQL v najglobljem nivoju najprej preračuna zadnji datum akcije s težo (relevanten datum, ki spremeni izračun izdelanih kosov) ali v primeru neobstoječega datuma kot varovalo časovnosti preračunavanja nad celotno podatkovno tabelo prevzame najnižjo datumsko mejo iz nastavitvev. Nato algoritem poveže set modelov preko parametrov s proizvodno linijo ter vsemi tehničnimi kodami, ki so vezane na specifičen set modelov, ter preračuna število izdelanih kosov z uporabo tabel obstoječega ERP sistema. Na koncu poizvedba SQL kot rezultat vrne tabelo vseh aktivnih setov modelov z njihovimi mejami življenja, preračunom izdelanih kosov ter osnovnimi podatki o setu modelov.

3.4.3 Prikaz stanja življenja modelov

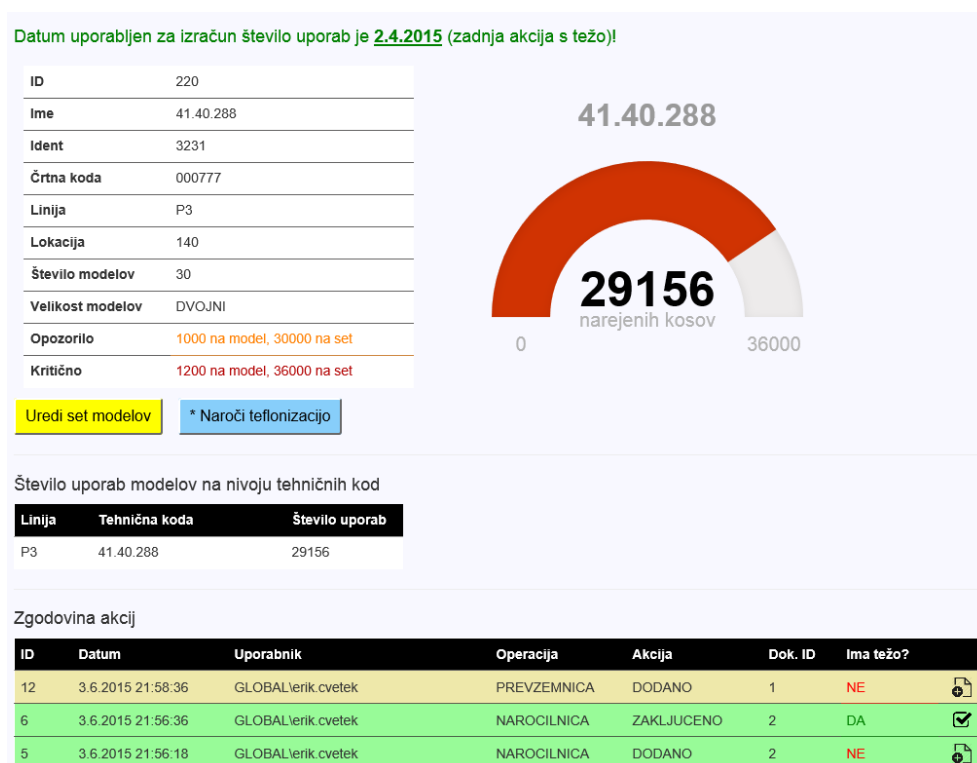
Prikaz stanja življenja modelov je za uporabnika najpomembnejši prikaz indikacije in mora biti čimbolj pregleden ter grafično enostaven. Pri implementaciji prikaza smo uporabili brezplačno odprtokodno knjižnico JustGage. Na sliki 3.3 je prikazan tipični pogled pregleda nad stanjem življenja s filtriranjem po proizvodnih linijah. V zaledju je za prikaz podatkov uporabljena poizvedba SQL, ki je opisana pod točko 3.4.2. S klikom na set modelov nas aplikacija preusmeri na detajlni pogled, podrobneje opisan pod točko 3.4.4.

3.4.4 Detajli seta modelov

Ker je za uporabnika občasno pomembna tudi analiza podatkov, ki so uporabljeni pri izračunu statusa življenja, smo v ta namen ustvarili stran s prikazom detajlov ter zgodovine akcij nad izbranim setom modelov. Na sliki 3.4 lahko vidimo primer prikaza detajlov za izbran set modelov. Na vrhu na levi strani so uporabniku prikazani osnovni podatki o setu modelov oziroma



Slika 3.3: Prikaz stanja življenja modelov.



Slika 3.4: Prikaz detajlov izračuna z zgodovino akcij seta modelov.

osnovne nastavitve, na vrhu na desni strani pa števec življenja. Pod osnovnimi podatki se nahajata gumba za hitro preusmeritev na naročilnico oziroma na urejanje seta modelov.

Na osrednjem delu strani se nahaja podrobna analiza izračuna izdelanih oziroma narejenih kosov, ki so prikazani na števcu. V tem delu uporabnik lahko vidi število kosov izdelanih na posamezni proizvodni liniji pod posamezno tehnično kodo. Ker je lahko set modelov vezan na eno ali več tehničnih kod, je ta del analize uporaben le v primeru dveh ali več vezav tehnične kode, saj je v primeru vezave na le eno tehnično kodo izračun narejen po principu ena proti ena.

Na spodnjem delu strani lahko uporabnik vidi celotno zgodovino akcij nad izbranim setom z modeli. V tem delu je prikazano vse od osnovnih modifikacij nastavitve do naročilnic ter prevzemnic. Ta del uporabniku služi

tudi za odkrivanje napak v primerih napačnih nastavitev ter prezgodnjih ali prepoznih servisov teflonskih oblog.

3.4.5 Naročilnice

Pri implementaciji naročilnic smo proces kreacije naročilnice zaradi lažjega razumevanja razdelili na tri korake. V prvem koraku uporabniku ponudimo osnoven nabor polj naročilnice, in sicer številko naročilnice (preddefinirano polje z onemogočenim spreminjanjem), datum izdaje in kraj izdaje (preddefinirani polji z omogočenimi spremembami uporabnika), dobavitelj (spustni seznam vseh dobaviteljev) ter stroškovno mesto (uporabnikov vnos). S klikom na naslednji korak se v aplikaciji avtomatsko kreira naročilnica s statusom *IZVEDBI*. Uporabnik lahko naročilnico ob možnem preklicu procesa kasneje zaključi ali izbriše. V drugem koraku uporabnik s seznamom vseh aktivnih setov modelov izbere tiste, ki jih bo poslal na servis teflonskih oblog. V tretjem, zadnjem koraku, uporabnik lahko popravi število modelov za pošiljanje na servis in s klikom na zaključek zaključi naročilnico. Na sliki 3.5 je prikazan zadnji korak, ob katerem se uporabniku odpre novo okno z izdelano naročilnico primerno za tiskanje. S tem korakom tudi aplikacija v ozadju avtomatsko spremeni status naročilnice v *ZAKLJUČENO*.

3.4.6 Prezemnice

Ker je uporabniku pomembno tudi vodenje evidence odsotnosti in sledenje razpoložljivosti setov z modeli, smo v ta namen implementirali sistem prevzemnic. Ko uporabnik ustvari naročilnico in odda set modelov na servis teflonskih oblog, se v aplikaciji pod oddelkom **PREVZEMNICE** prikaže oddana naročilnica s seti modelov, ki so trenutno na servisu in so pripravljeni na prevzem nazaj v prostore podjetja. Tako lahko uporabnik enostavno in hitro ob prevzemu naredi prevzemnico in zaključi celoten proces naročila. S tem korakom aplikacija v ozadju zaklene vezano naročilnico in uporabniku onemogoči brisanje brez predhodno izbrisane vezane prevzemnice. S tem v



ŠT. NAROČILNICE: 0005/2015

Naročnik: Sogefi filtration d.o.o., Ladja 11, Medvode, SI

Zavezanec za DDV: da
DDV številka: SI94321850

Datum naročila: 12.06.2015
Kraj: Ladja, Medvode, Slovenija

Dobavitelj: Obnavljanje teflonskih oblog, Janez Novak d.o.o., Medvode, Slovenija
Stroškovno mesto: 1133

Št.	Količina	EM	Velikost	Tip	Set modelov
1	64	kos	DVOJNI	NIZKI	Set modelov 41.40.111A
2	67	kos	DVOJNI	VISOKI	Set modelov 41.40.111A
3	1	kos	DVOJNI	NIZKI	Set modelov 41.40.303
4	1	kos	DVOJNI	NIZKI	Set modelov 41.40.328
5	18	kos	vsi	vsi	VARJENJE MODELOV

Naročil: <i>Medvode Maintenance</i>	Nadrejeni:	Direktor operacij:	Finančni direktor:	Generalni direktor ali namestnik:
Podpis:	Podpis:	Podpis:	Podpis:	Podpis:

Slika 3.5: Prikaz zaključene naročilnice pripravljene na tiskanje in podpisovanje.

Prejeta sporočila za pregled iz proizvodnje					
ID	Datum in ura oddaje	Lokacija	Oddal	Set modelov	Sporočilo
9	16.6.2015 20:36:54	P2	Demi Moore	41.40.292	modeli so nujno potrebni teflonizacije in naročiti dodatnih 20 modelov za polno delovanje linije še posebej ker je stekla proizvodnja za GMIH
14	22.6.2015 12:52:08	P4	Scarlett Johansson	41.40.344	Nujna teflonizacija. PUR se prijema na modele (poškodovani filtri).
15	24.6.2015 15:59:32	P3	Keira Knightley	41.40.401	NUJNO POTREBNO TEFLONIZIRATI MODELE KOMPLET
123					
Seznam vseh pregledov					
<input type="checkbox"/> Pokaži izbrisano					
ID	Datum in ura pregleda	ID spor.	Pregledal	Set modelov	Komentar
10	26.6.2015 13:29:16	13	GLOBAL\medvode.maintenance	41.40.254	MODELI SO PRIVIJACENI , PREMALO PROSTORA ZA VZMETNE PODLOŠKE
9	19.6.2015 8:51:23	12	GLOBAL\medvode.maintenance	41.40.320	ODŠLI NA TEFLONIZACIJO
8	19.6.2015 6:12:58	10	GLOBAL\medvode.maintenance	41.40.372	ODŠLI NA TEFLONIZACIJO
7	19.6.2015 6:10:19	8	GLOBAL\medvode.maintenance	41.40.319	ODŠLI NA TEFLONIZACIJO
5	12.6.2015 6:18:40	7	GLOBAL\medvode.maintenance	41.40.111A	poravnani in odšli na teflonizacijo
12					

Slika 3.6: Prikaz čakajočih sporočil in seznam pregledov.

aplikaciji zagotovimo celovitost ter strukturo referenčnih podatkov oziroma izvedenih akcij.

3.4.7 Pregledi

Na sliki 3.6 vidimo okno s pregledi in čakalno listo sporočil poslanih iz namizne aplikacije proizvodnih računalnikov. Uporabniki v vzdrževanju lahko na podlagi sporočil vnesejo pregled seta z modeli, to pa se v seznamu beleži kot pregled na zahtevo. Ker pa pregled ni vedno začet le na zahtevo, lahko uporabniki v vzdrževanju vnesejo tudi preglede brez predhodne podlage sporočila. Taki pregledi se smatrajo kot preventivni pregledi kakovosti.

Spletni dokument pregledov je tako razdeljen na dva dela, zgodnji del s čakajočo listo sporočil, urejena po FIFO (angl. *First in, first out*) sistemu, ter spodnji del, ki prikazuje seznam zgodovine pregledov.

3.4.8 Poročila

Poročila so zelo pomemben vidik celotnega sistema, saj lahko na podlagi le-teh različni oddelki podjetja opravijo analizo pretekle prakse dela setov z modeli ter na podlagi pridobljenih podatkov načrtujejo izboljšave kakovosti v prihodnjem mesecu. Poročila so implementirana z uporabo sistema za poročanja SSRS [25, 26]. Prikazani podatki so bazirani direktno z uporabo poizvedb SQL klicanih nad podatkovno bazo. Določen del poročil je tudi datumsko planiran s sistemom avtomatskega poročanja na dnevni, tedenski in mesečni ravni z uporabo elektronskega sporočanja po skupinah ključnih uporabnikov. S tem smo zagotovili tekoče obveščanje o uspešnosti ter stanju v določeni točki mesečnega napredka.

3.4.9 Dnevnik uporabe

Ker je dnevnik uporabe ponavadi vedno prvo orodje, na katerega se uporabniki obrnejo, ko iščejo vzrok napake ali le napako v podatkih, smo v spletno aplikacijo implementirali enostaven pregled zgodovine akcij izvedenih v aplikaciji. V zgodovini aktivnosti hranimo in prikazujemo: kdo ter kdaj je izvedel določeno akcijo v določenem oddelku aplikacije, identifikacijsko številko dokumenta ter ali je imela akcija težo pri pomembnih izračunih oziroma ali je šlo le za evidenčno operacijo nad elementi znotraj spletne aplikacije.

Na sliki 3.7 je prikazan primer prikaza zgodovine aktivnosti uporabe spletne aplikacije.

Zgodovina aktivnosti							
ID	Datum	Uporabnik	Operacija	Akcija	Dok. ID	Ima težo?	
65	30.6.2015 10:03:06	GLOBAL\medvode.maintenance	SPOROČILO	IZBRISANO	17	NE	
64	29.6.2015 6:53:03	GLOBAL\medvode.maintenance	PREVZEMNICA	DODANO	3	NE	
63	29.6.2015 6:51:28	GLOBAL\medvode.maintenance	PREVZEMNICA	DODANO	2	NE	
62	29.6.2015 6:50:42	GLOBAL\medvode.maintenance	PREVZEMNICA	DODANO	1	NE	
61	26.6.2015 13:29:16	GLOBAL\medvode.maintenance	PREGLED	DODANO	10	NE	
60	26.6.2015 11:42:46	GLOBAL\medvode.maintenance	SPOROČILO	IZBRISANO	16	NE	
59	19.6.2015 15:34:04	GLOBAL\erik.cvetek	NAROCILNICA	IZBRISANO	17	DA	
58	19.6.2015 15:33:57	GLOBAL\erik.cvetek	NAROCILNICA	DODANO	17	NE	
57	19.6.2015 14:03:10	GLOBAL\erik.cvetek	NAROCILNICA	IZBRISANO	10	DA	
56	19.6.2015 8:53:58	GLOBAL\medvode.maintenance	NAROCILNICA	ZAKLJUCENO	16	DA	
55	19.6.2015 8:52:40	GLOBAL\medvode.maintenance	NAROCILNICA	DODANO	16	NE	
54	19.6.2015 8:51:23	GLOBAL\medvode.maintenance	PREGLED	DODANO	9	NE	
53	19.6.2015 8:50:24	GLOBAL\medvode.maintenance	SPOROČILO	IZBRISANO	11	NE	
52	19.6.2015 8:50:11	GLOBAL\medvode.maintenance	SPOROČILO	IZBRISANO	11	NE	
51	19.6.2015 7:14:44	GLOBAL\medvode.maintenance	NAROCILNICA	ZAKLJUCENO	15	DA	
12345							

Slika 3.7: Prikaz zgodovine aktivnosti uporabe spletne aplikacije.

3.5 Implementacija namizne aplikacije

Že pri načrtovanju aplikacije smo se zavedali potencialne slabosti nadzora kvalitete pri velikem številu setov z modeli, zato smo se odločili implementirati sistem sporočanja med zaposlenimi na proizvodnih linijah in uporabnikom spletne aplikacije. V ta namen smo razvili enostavno namizno aplikacijo, preko katere operaterji proizvodnih linij lahko pošljejo sporočilo v spletno aplikacijo. Tako smo razbremenili tiste zaposlene, ki so odgovorni za vzdrževanje ter pregledovanje teflonskih oblog modelov.

3.5.1 Prijava

Pri implementaciji namizne aplikacije smo hoteli zagotoviti tudi sledljivost sporočil, zato smo v aplikacijo vključili sistem prijave. Prijavo smo integrirali nad že obstoječ informacijski ERP sistem ter se tako izognili dvojni nepotrebnim administraciji uporabnikov. Tako smo tudi zmanjšali uporabo doda-

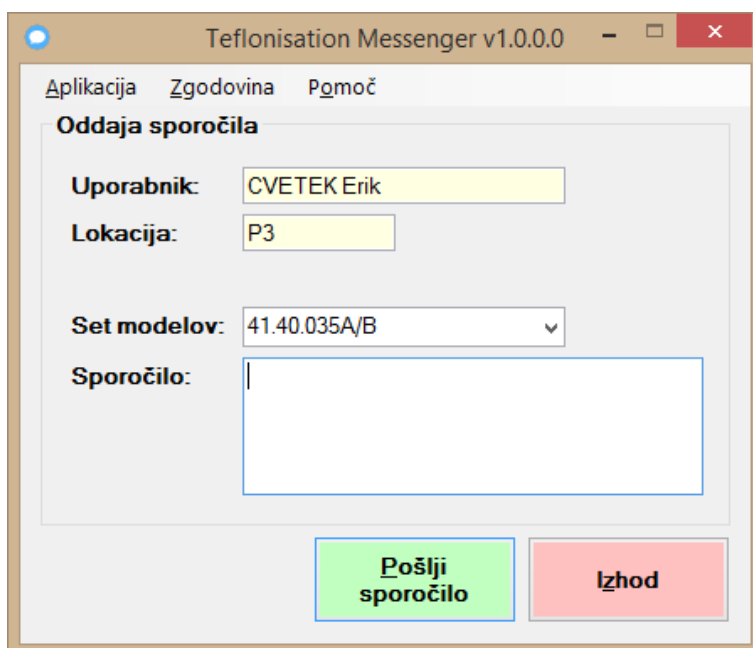
Slika 3.8: Prikaz prijavnega obrazca v namizno aplikacijo z izborom lokacije.

tne aplikacije ter proizvodnim uporabnikom zagotovili transparenten pristop k dodatku sistema.

Na sliki 3.8 je prikazano prijavno okno v aplikacijo. Pri prijavi smo uporabnikom omogočili izbor lokacije proizvodne linije, a le tistimi, ki uporabljajo računalnik v skupni rabi na več proizvodnih linijah. Izbor lokacije je voden preko tabele v podatkovni bazi MSSQL. Če se v tabeli nahaja polno omrežno ime računalnika, potem aplikacija avtomatsko zaklene izborna okno z izborom shranjenim v tabeli. Na sliki 3.9 je prikazana tabela s podatki o računalnikih ter njihovih lokacijah. Če uporabnik uporablja računalnik, ki mrežnega imena nima v tabeli, aplikacija dovoli uporabniku ročni izbor lokacije.

	ID_CLIENT	ADCN	DEFAULT_LOCATION
1	1	SI-MED-PC-0049	V1
2	2	SI-MED-PC-0038	P3
3	3	SI-MED-PC-0083	P2
4	4	SI-MED-PC-0112	P4

Slika 3.9: Prikaz tabele predizbranih lokacij računalnikov.



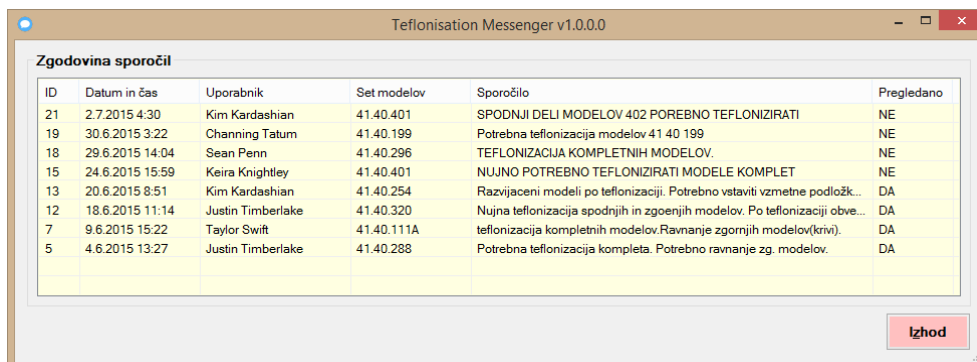
Slika 3.10: Prikaz glavnega okna namizne aplikacije.

3.5.2 Glavno okno

Pri implementaciji glavnega okna namizne aplikacije smo se čim bolj poskušali približati minimalističnem načinu uporabe in izgleda ter poskusili uporabniku zagotoviti čimbolj transparentno in hitro uporabo. Na sliki 3.10 je prikazano glavno okno aplikacije. V zgornjem delu prikažemo prijavljenega uporabnika in lokacijo, s katere je prijavljen, v spodnjem delu pa uporabnik sam izbere set modelov, za katerega pošilja sporočilo ter vpiše telo sporočila. V izboru setov z modeli so prikazani samo tisti seti modelov, ki so neposredno vezani na izbrano lokacijo v nastavitvah spletne aplikacije. S tem smo znižali tudi mejo verjetnosti uporabniške napake napačnega izbora seta z modeli.

3.5.3 Zgodovina

Proizvodnim uporabnikom smo omogočili tudi pregled zgodovine vseh poslanih sporočil z njihove lokacije ter njihovega statusa. Na sliki 3.11 je prikazano



ID	Datum in čas	Uporabnik	Set modelov	Sporočilo	Pregledano
21	2.7.2015 4:30	Kim Kardashian	41.40.401	SPODNJI DELI MODELOV 402 POREBNO TEFLONIZIRATI	NE
19	30.6.2015 3:22	Channing Tatum	41.40.199	Potrebna teflonizacija modelov 41 40 199	NE
18	29.6.2015 14:04	Sean Penn	41.40.296	TEFLONIZACIJA KOMPLETNIH MODELOV.	NE
15	24.6.2015 15:59	Keira Knightley	41.40.401	NUJNO POTREBNO TEFLONIZIRATI MODELE KOMPLET	NE
13	20.6.2015 8:51	Kim Kardashian	41.40.254	Razvijaceni modeli po teflonizaciji. Potrebno vstaviti vzmetne podložk...	DA
12	18.6.2015 11:14	Justin Timberlake	41.40.320	Nujna teflonizacija spodnjih in zgoenjih modelov. Po teflonizaciji obve...	DA
7	9.6.2015 15:22	Taylor Swift	41.40.111A	teflonizacija kompletnih modelov.Ravnanje zgornjih modelov(krivi).	DA
5	4.6.2015 13:27	Justin Timberlake	41.40.288	Potrebna teflonizacija kompleta. Potrebno ravnanje zg. modelov.	DA

Slika 3.11: Prikaz okna z zgodovino sporočil.

okno zgodovine sporočil. Na desni strani okna lahko uporabniki aplikacije vidijo status pregleda. Ko odgovorni za vzdrževanje modelov opravi pregled seta z modeli in dejanje zabeleži v spletni aplikaciji pod rubriko 3.4.7, se spremeni tudi indikator v stolpcu *Pregledano*.

Poglavje 4

Sklepne ugotovitve

V prvem delu projekta je bila naša osredotočenost ter večina časa usmerjena k načrtovanju aplikacije ter izboru uporabljenih tehnologij. V tej fazi projekta smo se kar nekajkrat sestali z vodji oddelkov ter temeljito predebatirali funkcionalnosti ter ideje implementacije, kar se nam je kasneje eksponentno obrestovalo pri porabljenem času za implementacijo. Aplikacijo smo sproti po modulih vpeljevali v produkcijo, kjer so jo vzporedno z razvojem testirali kar naši končni uporabniki.

Med implementacijo ter sprotnim razvojem smo dobili tudi veliko idej, kako bi lahko celoten sistem še dodatno izboljšali, zato smo določene segmente idej kar sproti vpeljevali v sistem. Vsak set modelov tako dodatno nosi še črtno kodo in identifikacijsko številko, kar bi bilo lahko kasneje uporabljeno za lažjo ter bistveno hitrejšo identifikacijo orodja z uporabo čitalnika črtnih kod, ki so že na lokacijah poleg vsakega računalnika v proizvodnih obratih. V sistemu bi lahko poleg orodja, ki potrebuje servis teflonskih oblog, vpeljali tudi potrošno orodje primerno za izdelavo točno predpisanem številu kosov, preden se zavrže. Pri procesu tiskanja naročilnic pa smo sistem že pripravili na naslednji korak, kjer bi eliminirali dejansko tiskanje na papir ter vpeljali elektronsko podpisovanje z uporabo veljavnih certifikatov znotraj podjetja.

Sistem vodenja orodja je bil v celoti predan v produkcijo na področju

slovenske regije, načrtuje pa se že generalizacija in regijska lokalizacija celotnega sistema za implementacijo po vseh ostalih proizvodnih obratih po svetu. Starševsko podjetje Sogefi Group namreč obsega kar 42 proizvodnih obratov, ki so locirani v 21 državah na štirih kontinentih sveta. Vsi generalni direktorji sestrskih podjetij so že bili seznanjeni z našo rešitvijo upravljanja obnovljivega in potrošenega orodja. Pokazali so interes za možno implementacijo orodja na njihovih lokacijah.

Kljub številnim idejam za izboljšavo ter optimizacijo ocenjujemo, da je bil projekt v splošnem zelo uspešen in primeren zahtevam ter ciljem diplomskega dela.

Literatura

- [1] HTML. [Online]. Dosegljivo:
<http://en.wikipedia.org/wiki/HTML>. [Dostopano 28. 6. 2015].
- [2] Dave Raggett, Arnaud Le Hors, Ian Jacobs, et al. Html 4.01 specification. *W3C recommendation*, 24, 1999.
- [3] Ian S Graham. *The HTML sourcebook*. John Wiley & Sons, Inc., 1995.
- [4] CSS. [Online]. Dosegljivo:
http://en.wikipedia.org/wiki/Cascading_Style_Sheets. [Dostopano 28. 6. 2015].
- [5] CSS. [Online]. Dosegljivo:
<http://sl.wikipedia.org/wiki/CSS>. [Dostopano 28. 6. 2015].
- [6] C#. [Online]. Dosegljivo:
[http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)). [Dostopano 28. 6. 2015].
- [7] A. Hejlsberg, S. Wiltamuth, and P. Golde. *The C# Programming Language*. Microsoft .NET development series. Addison-Wesley, 2006.
- [8] J. Liberty. *Programming C#: Building .NET Applications with C#*. O'Reilly Media, 2005.
- [9] JavaScript. [Online]. Dosegljivo:
<http://en.wikipedia.org/wiki/JavaScript>. [Dostopano 28. 6. 2015].

- [10] JavaScript. [Online]. Dosegljivo:
<http://sl.wikipedia.org/wiki/JavaScript>. [Dostopano 28. 6. 2015].
- [11] Bootstrap WYSIHTML5. [Online]. Dosegljivo:
<http://github.com/xing/wysihtml5>. [Dostopano 28. 6. 2015].
- [12] Bootstrap WYSIHTML5. [Online]. Dosegljivo:
[http://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)). [Dostopano 28. 6. 2015].
- [13] Microsoft Visual Studio.[Online]. Dosegljivo:
<http://www.visualstudio.com>. [Dostopano 28. 6. 2015].
- [14] Microsoft Visual Studio. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Dostopano 28. 6. 2015].
- [15] Sam Guckenheimer and Juan J. Perez. *Software Engineering with Microsoft Visual Studio Team System (Microsoft .NET Development Series)*. Addison-Wesley Professional, 2006.
- [16] Brian Johnson, Craig Skibo, and Marc Young. *Inside Visual Studio. NET*. Microsoft Press, 2003.
- [17] JustGage. [Online]. Dosegljivo:
<http://justgage.com>. [Dostopano 28. 6. 2015].
- [18] Microsoft SQL Server.[Online]. Dosegljivo:
http://en.wikipedia.org/wiki/Microsoft_SQL_Server. [Dostopano 28. 6. 2015].
- [19] Microsoft SQL Server. [Online]. Dosegljivo:
<http://www.microsoft.com/en-us/server-cloud/products/sql-server>. [Dostopano 28. 6. 2015].

-
- [20] SQL Server Management Studio. [Online]. Dosegljivo:
http://en.wikipedia.org/wiki/SQL_Server_Management_Studio. [Dostopano 28. 6. 2015].
- [21] ASP.NET. [Online]. Dosegljivo:
<http://en.wikipedia.org/wiki/ASP.NET>. [Dostopano 28. 6. 2015].
- [22] ASP.NET. [Online]. Dosegljivo:
<http://www.asp.net>. [Dostopano 28. 6. 2015].
- [23] J. Liberty and D. Hurwitz. *Programming ASP.NET*. Building web applications and services using C# and VB.NET. O'Reilly, 2003.
- [24] D. Esposito. *Programming Microsoft ASP.NET 4*. Developer Reference. Pearson Education, 2011.
- [25] SQL Server Reporting Services. [Online]. Dosegljivo:
http://en.wikipedia.org/wiki/SQL_Server_Reporting_Services. [Dostopano 28. 6. 2015].
- [26] Brian Larson. *Microsoft SQL Server 2005 Reporting Services*. McGraw-Hill, Inc., New York, NY, USA, 2 edition, 2006.

Priloga A

Izsek poizvedbe SQL algoritma
za izračun stanja življenja setov
z modeli

```

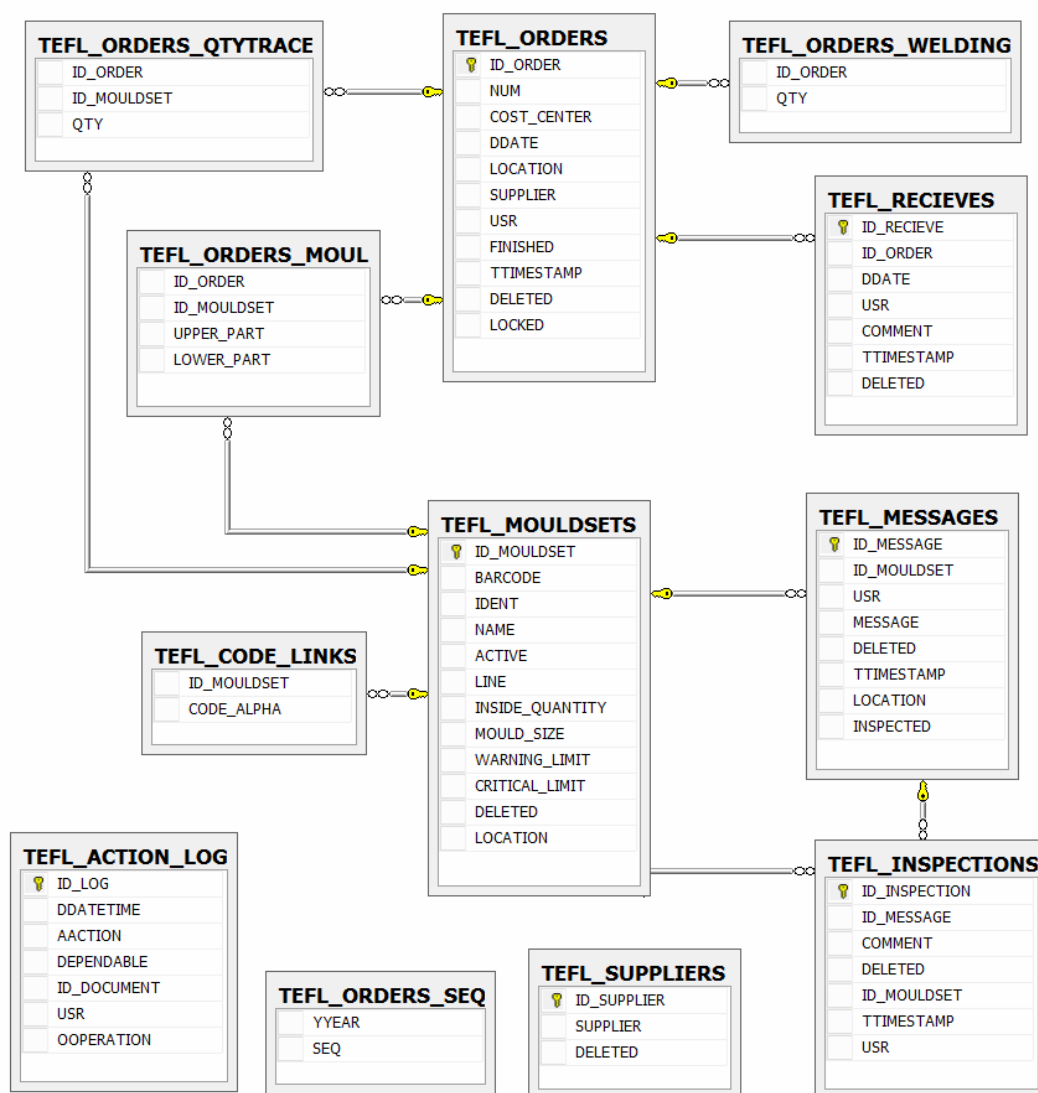
SELECT X.*, ISNULL(Y.DDATE, @p2) AS DDATE,
  (SELECT ISNULL(SUM(Q.BONNE), 0) AS QTY
   FROM
    (
      SELECT SUM(P.QTE_BONNE) AS BONNE, A.CODE_ALPHA
      FROM SP_HISTO_PROD AS P INNER JOIN
        SP_PARA_ARTICLES AS A ON P.REF_ART = A.REF_ART
      WHERE
        P.JOUR > (
          ISNULL(Y.DDATE, @p2)
        )
      AND A.CODE_ALPHA IN (
        SELECT CODE_ALPHA
        FROM TEFL_CODE_LINKS AS TCL
        WHERE TCL.ID_MOULDSET = X.ID_MOULDSET
      )
      AND P.CHAINE IN (
        SELECT LINE COLLATE French_CI_AS
        FROM TEFL_MOULDSETS AS MLD
        WHERE MLD.ID_MOULDSET = X.ID_MOULDSET
      )
      GROUP BY A.CODE_ALPHA
    ) AS Q
  ) AS QTY
FROM
  (
    SELECT ID_MOULDSET, BARCODE, IDENT, NAME, ACTIVE, WARNING_LIMIT, CRITICAL_LIMIT,
      INSIDE_QUANTITY, LINE
    FROM TEFL_MOULDSETS
    WHERE ACTIVE = 1 AND DELETED = 0
  ) AS X
FULL OUTER JOIN
  (
    SELECT MAX(O.DDATE) AS DDATE, M.ID_MOULDSET
    FROM TEFL_ORDERS_MOULDS AS M INNER JOIN
      TEFL_ORDERS AS O ON M.ID_ORDER = O.ID_ORDER INNER JOIN
      TEFL_CODE_LINKS AS L ON M.ID_MOULDSET = L.ID_MOULDSET
    WHERE O.DELETED = 0 AND O.FINISHED = 1
    GROUP BY M.ID_MOULDSET
  ) AS Y
ON X.ID_MOULDSET = Y.ID_MOULDSET
WHERE X.LINE LIKE @p1
ORDER BY X.NAME ASC

```

Slika A.1: Izsek poizvedbe SQL algoritma.

Priloga B

Model podatkovne baze



Slika B.1: Prikaz konceptualnega diagrama podatkovne baze.